REMARKS

The above amendment and these remarks are responsive to the communication from Examiner Dustin Nguyen dated 3/9/2005.

Claims 1-27 are in the case, none as yet allowed.

35 U.S.C. 112

Claims 12 and 15 have been rejected under 35 U.S.C. 112, second paragraph, as being indefinite, and claim 28 for being incomplete.

Applicants have amended claims 12 and 15 to clarify the claim, and canceled claim 28.

35 U.S.C. 103

Claims 1-7, 9-28 have been rejected under 35 U.S.C. 103(a) over Butts et al. [US Patent 6,233,543, hereinafter

Butts] in view of Faybishenko [US Patent 5,757,925].

With respect to claim 1, Butts simply describes communicating to a 1/2 duplex block mode server (Fig. 1, box 18), in 1/2 duplex block mode, and to a full duplex character interactive I/O server (Fig. 1, box 19) in full duplex character interactive I/O mode. He simply supports both modes, just as do most good Telnet client emulators.

However, at no point does Butts describe communicating to a full duplex character interactive I/O server over a 1/2 duplex block mode interface. Referring to Fig. 1, note that Butts shows a separate interfaces 30 to box 19, the full duplex server, and to box 18, the 1/2 duplex server.

Butts columns 8 through 26 discusses all protocols he supports (3270, 5250, 3287, VT220) for his emulation protocol. As such, these columns contain commands and definitions for protocols and fields that are valid in some modes but not relevant in others. Specifically, Col. 10, lines 3-13, describe a command that would only be used in ASCII (VTxx) mode, not in 1/2 duplex block mode, as the amended claim requires.

With respect to claim 2, Faybishenko describes a server/client graphical user interface (GUI) which minimizes latency time and data transmission between a client and a server by causing the to handle locally whatever GUI events it can during the execution of an application.

In Col. 9, Faybishenko describes how buffering of keystroke events occur until some event forces the buffer to be cleared and transmitted to the server. This buffering of keystroke data is exactly what transpires normally when communicating with a 1/2 duplex block mode server.

However, in Applicants' invention, a method is described of sending each keystrike individually (that is, full duplex character interactive mode) to the server as they occur, even though the end client is forced to communicate in 1/2 duplex block mode by the (2370 or 5250) mode enforced by the initial server.

The buffer described by Faybishenko is not auto-enter, as applicants' claim 2 specifically requires; that is, Faybishenko requires an event, such as a button press, button release, enter key, or the like, to cause to be sent), and it is echoed locally to the display screen (i.e.,

it is not a non-display entry).

With respect to claim 3, Faybishenko describes [Abstract; and Col. 12, lines 1-6] a security proxy which can act as an intermediary between the client and security server, thereby effectively making the application server, security server, and database server hidden and inaccessible to outside users. As such, Faybishenko is describing how the servers can be hidden, not a client keystroke buffer. It makes no mention of the client keystroke buffer, and is therefore not relevant to Applicants' claim 3.

With respect to claims 4 and 5, Applicants' invention distinguishes as previously discussed with respect to claim 1. Further, Butts Col. 12 is a continuation of the Butts emulation protocol specification, and is meant to be general since it supports all emulation modes. Nowhere is there a description of a one byte character input field that has auto-enter and non-display attributes when operating in 1/2 duplex block mode as specified by Applicants' claims 4 and 5.

With respect to claim 6, Applicants argue that Butts does not describe operating in cascaded mode at all.

Referring to Butts, Fig. 1, a separate interface connection 30 is provided to each server type that is supported. If Butts were operating in cascaded mode, there would be an interface line drawn between box 18 and box 19, depicting the cascaded connection available between the 1/2 duplex block mode server (box 18, which would then be the cascaded client), and the full duplex character interactive I/O server (box 19).

With respect to claim 7, see the arguments presented with respect to claims 1 and 4, above. Col. 19 is a continuation of the Butts emulation protocol specification, and is meant to be general since it purports to support all emulation modes.

With respect to claims 9 and 10, see the arguments presented above with respect to claims 4 and 5, above.

With respect to claim 11, in Faybishenko the client keystrokes are locally echoed as they are typed and are transmitted client to server, not the other way around as Applicant's claim requires. The Faybishenko server responds with the results; that is, some events and new data [Col. 9, lines 13-22].

With respect to claim 12, in Faybishenko the client takes care of echoing characters to a text box, for example, in a screen in the proper place, in the proper font [Faybishenko, Col. 9, lines 22-26]. Fabishenko does not describe not echoing the character nor replacement of the echoed character.

With respect to claims 13-25, rejected for similar reasons as stated for claims 1-12, see Applicants' arguments distinguishing claims 1-12, above.

With respect to claim 26 and 27, Butts does not describe cascaded Telnet at all. All Butts' transfers are from Telnet client to Telnet server. There is no discussion of any data transfers between a Telnet client and an additional Telnet client application running on the Telnet server cascaded to a Unix server. See also the discussion above with respect to claim 6.

Claim 28 has been canceled.

Claim 8 has been rejected under 35 U.S.C. 103(a) over Butts, in view of Faybishenko, and further in view of

Shoquist et al. [U.S. Patent 5,361,199, hereinafter Shoquist].

Claim 8 depends from claim 4, and is distinguished from Butts and Faybishenko as previously described. Applicants did not invent converting a character from ASCII to EBCDIC, but it is an important step in the multi-step process the subject of claim 8. Further, Shoquist does not run in either 1/2 duplex block mode or full duplex character interactive I/O mode. In fact, Shoquist does not use Telnet at all for communicating to the mainframe. As such, Shoquist in combination with Butts and Faybishenko does not teach the essential elements of Applicants' invention as set forth in claims 4 and 8.

SUMMARY AND CONCLUSION

Applicants urge that the above amendments be entered and the case passed to issue with claims 1-27.

The Application is believed to be in condition for allowance and such action by the Examiner is urged. Should differences remain, however, which do not place one/more of

the remaining claims in condition for allowance, the Examiner is requested to phone the undersigned at the number provided below for the purpose of providing constructive assistance and suggestions in accordance with M.P.E.P. Sections 707.02(j) and 707.03 in order that allowable claims can be presented, thereby placing the Application in condition for allowance without further proceedings being necessary.

Sincerely,

Richard G. Hartmann, et al.

By

Shelley M Beckstrand
Reg. No. 24,886

Date: 24 May 2005

Shelley M Beckstrand, P.C.
Attorney at Law
61 Glenmont Road
Woodlawn, VA 24381-1341

Phone: (276) 238-1972
Fax: (276) 238-1545